



(11) Publication number : **0 650 144 A1**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : **94307833.7**

(51) Int. Cl.⁶ : **G06T 3/00**

(22) Date of filing : **25.10.94**

(30) Priority : **26.10.93 US 143323**

(43) Date of publication of application :
26.04.95 Bulletin 95/17

(84) Designated Contracting States :
DE FR GB

(71) Applicant : **XEROX CORPORATION**
Xerox Square
Rochester New York 14644 (US)

(72) Inventor : **Robertson, George G.**
641 Greenwich Lane
Foster City CA 94404 (US)
Inventor : **Mackinlay, Jock D.**
3240 Ross Road
Palo Alto CA 94303 (US)

(74) Representative : **Reynolds, Julian David et al**
Rank Xerox Ltd
Patent Department
Parkway
Marlow Buckinghamshire SL7 1YL (GB)

(54) **Image lens.**

(57) A method for displaying an image (200) on a display surface which preserves context and detail information when the image is compressed to fit onto a display surface. The image (200) is transformed from a plane onto a truncated pyramid (203), and then a three-dimensional perspective projection (Fig.4(b)) of the pyramid (203) is taken. This perspective provides a detailed view of portions of the image underlying an image lens panel (202), and context view of remaining portions of the image (200), which underlie side panels. The side panels are rotated and expanded to fill the remainder of the view after the detailed image under the image lens (202) is expanded.

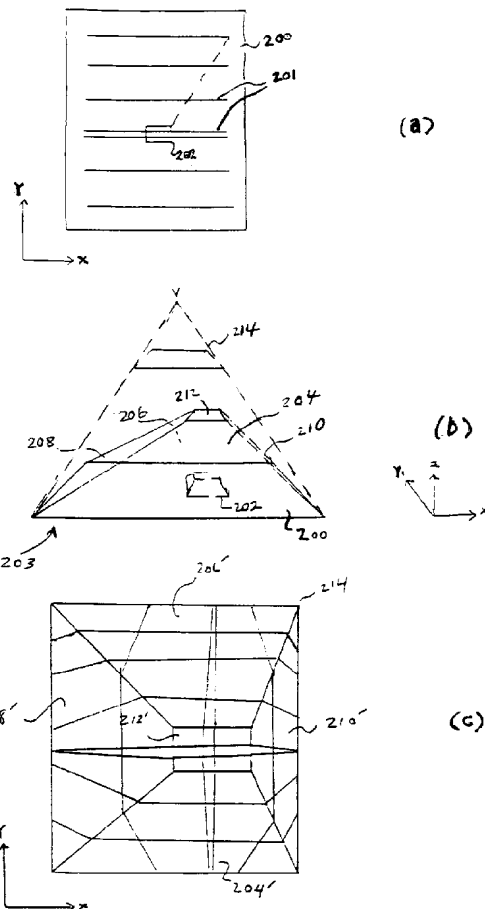


FIG. 4

The present invention relates to the field of visual displays of information. More specifically, the invention solves the problem of how to display an image containing large amounts of information on a display where the image is too large to be displayed in its entirety at full resolution.

For the purposes of this application, several variations on an image will be defined. An image is a visual display of information, or a logical representation of what is to become a visual display of information. An image could be, but is not limited to, a text document, a map, or a graph. The underlying information contained in an image depends on its form. For example, information in a text document includes the information content of its words, the information content of the positions of the words within the document, and the information content of text formatting and fonts, etc. Information has many aspects, but two aspects are of particular concern here: detail and context. Detail information in a text document image indicates which local information such as which fonts are used and which characters are present in a local area of the image, while detail information in a map indicates the symbols used on the map and the path taken by a road (in the case of road maps). Context information, on the other hand, places these details in perspective with the other information of the total image.

An image standing alone, by definition, contains all the information content of that image. When an image contains too much information to be displayed all at once on a given display device, a substitute image comprising less than all the information content of the "full" image must be accepted. The term "full image" will be used herein to refer to an original image, before any information-losing operations are applied to the image to form a displayable substitute image. Of course, such a substitute image could just as well be a full image for a yet lower resolution display, however for simplicity, full images will be assumed to be ideal in that they contain all the information necessary to display the full image on an ideal display (i.e., infinite resolution, size, color, etc.).

When the full image is undisplayable in its entirety, the displayable image which is substituted for the full image is often either a detail image or a global image, or a combination of the two. A global image is the full image with resolution removed to allow the entire image to fit onto the display surface of the display device. Of course, the resolution might be so low that details are not available from this substituted image. A detail image shows the details, but only of a portion of the full image. With a detail image, the details of the image are available, but the global context of the details are lost. If a combination is used, the connection between the detail image and the global image will not be visually apparent, especially where the detailed image obscures more of the global image than is shown in the detailed image.

A display device or display system, as used herein, could be a computer monitor, printer, or the like, however computer monitors are more suitable for interactive displays of images. Where image processing is required, a display system will often include an image processor or other computer system. A display device might also be a device which behaves as if it displayed images but does not actually display the images. For example, a facsimile machine might manipulate images much the same manner as a computer display, but doesn't necessarily display an image, instead providing the image to another system which does display the provided image.

Due to the growth of computing power, users are demanding more and more information from computers, and want the information provided in a visually informative form. Take, for example, a business planner who is deciding where to locate a retail outlet. While it was at one time acceptable to work from a long print-out of numbers from a database showing population centers and the distances between them, now graphical two-dimensional ("2D") displays of the data overlaid on geographical data is expected. This output might be part of a geographical information system (GIS), which are well known.

The problem with such 2D displays is that a full image which is moderately complex cannot be displayed all at once with the detail necessary, due to the resolution of the information in the image and the resolution and size of the display surface. Often, a user must choose between a global image and a detail image. Generally, this global/context versus detail/focus tradeoff exists wherever context is important and the full image is moderately complex.

For example, if the full image is a multi-page document and locations of occurrences of a text string in the document are important, an image of the document might be displayed, with the occurrences highlighted. If the image of the entire document can be displayed and still be readable, detail and context are fully available. But, if the full image cannot be displayed, the user must often choose between an unreadable global image showing the context and readable detail images which lack global context information.

One partial solution is to display detail images over a global image, with the global image possibly including an indication of where the detail image fits into the full image. This approach is less than desirable, since the detail image is detached from the global context, and consequently a user viewing the images will need to jump back and forth visually between the global and detail images to observe both the details and the context of the information presented in the detail image. This would be quite difficult where many details, such as text or lines, extend beyond the edge of the detail image.

This problem is illustrated in Fig. 1. Fig. 1 shows three views of a full image 10, an image of a document containing the text of a sentence. Fig. 1(a) shows the full image, and the text in image 10 is readable (i.e., it displays all the information content of the document), but in any real application, the document to be imaged will be much longer and its text will have to be compressed to much smaller resolutions. In such a case, the text might not be completely readable (i.e., information embodied in the text will be lost) without magnification. Fig. 1(b) shows the effect of magnification of a region 13 of image 10 into a detail image 14, which is effected by magnifier 12. As a result of the magnification of region 13, other areas of image 10 are obscured, and the context of detail image 14 is lost due to the discontinuity between region 13 and image 10. Note that the discontinuity occurs at the edge of magnifier 12.

The obscuring and loss of context occurs with both physical magnifiers and logical magnifiers, and the cause is shown in Fig 1(c). Fig. 1(c) illustrates the view from a viewpoint 16 through magnifier 12 to region 13. Since magnifier 12, in effect, brings region 13 closer to viewpoint 16, the solid angle subtended by the advanced region is larger and corresponds to the bounds of an obscured region 18 in image 10.

US-A-4,790,028 describes an attempt to solve the problem of magnified areas obscuring global images, by using a "variably-scaled" display. In such a display, the full image is distorted such that the coordinates of an area of interest are expanded while the coordinates of the area just outside the area of interest is correspondingly compressed; the compressed portions and the remaining portions of the image are displayed with loss of information, with the loss being greater in the compressed area. However, this approach is computationally unwieldy, which results in slow response times to a user's request for amplification of detail, and is also unsuitable where the information content of the image includes text, which would be distorted to prevent easy reading, or includes characteristics of lines, such as their straightness, distances and relative angles of intersection. Furthermore, such a system does not take into effect the context information which may be preserved if compression is graduated from a lower compression near a detail image to a higher compression in more peripheral areas.

From the above it is seen that an improved system for displaying an image where both amplification of detail and context preservation of that detail within an image is needed.

The present invention provides a method for displaying an image on a display surface, preserving context and detail information even when the image contains too much detail to be displayed all at once on the display surface, comprising the steps of:

storing a data block representing objects appearing in the image, wherein a full image is an image generated from said data block without loss of information;

determining layout of said full image in two dimensions which is to be displayed on the display surface and includes context information;

dividing said layout into a lens panel which maps to portions of said full image containing desired detail information, and dividing a remainder of said layout not within said lens panel into a plurality of side panels;

mapping said lens panel and said plurality of side panels onto a truncated pyramid in a 3D space, said lens panel forming a truncated top surface of said truncated pyramid and said side panels forming wall surfaces of said truncated pyramid;

transforming said truncated pyramid through a perspective transform to form a 3D transformed layout, where a viewpoint for said perspective transform is above said lens panel; and

rendering said full image onto the display surface using said 3D transformed layout to form a rendered image, wherein said perspective transform provides a detailed view of said lens panel and context views of said plurality of side panels.

The method preferably further comprises the steps of (1) moving said viewpoint when said image lens would otherwise be out of view in a direction which results in said image lens appearing nearer to the center of the display surface, and/or (2) dividing said truncated top surface into a nested lens panel and four nested side panels, resulting in a transforming surface of a truncated pyramid atop a truncated pyramid.

Preferably image lens is positioned in a position relative to said viewpoint and said full image such that less than all of said plurality of side panels are visible on said display surface.

An improved display system which preserves context while providing detailed focus is provided by virtue of the present invention.

The invention further provides a context-preserving image display apparatus according to claim 6 of the appended claims.

The invention further provides a method of sealing an image, according to claim 9 of the appended claims.

In one embodiment of the present invention, a full image is presented as a lensed panel detail image and side panel images. The lensed panel contains a view of a portion (possibly all) of the full image to a desired resolution, surrounded by side panel images which show global context and are continuous with the lensed panel and adjacent side panel images. The maximum resolution of the panel images is determined by how

much display surface area is taken up by the lensed image, the location of the lensed image within the full image, and the area taken up by the full image. Taken together, the lensed panel image and the side panel images can occupy substantially all of a display surface with a global image containing all of the context of an full image, but which might leave off some detail in peripheral portions of the image.

5 In a specific embodiment, the full image is rectangular, the lensed panel is rectangular and parallel with the edges of the full image and the side panels are trapezoids. In this embodiment, the transformation of a full image into a lensed panel image and four panel images (top, bottom, left, and right), which together form a rectangle, is done by diving the full image into panels, mapping the panels onto a three-dimensional ("3D") truncated pyramid (the lensed panel is the top of the truncated pyramid) and then projecting the truncated pyramid into a viewing volume.

10 The viewing volume is a 3D perspective projection with respect to a viewpoint. The user first views the image as a global image flush with a constant Z plane, with the lensed panel lying under an image lens unmagnified. As the image lens is brought toward the viewer (i.e., the viewpoint; in a positive Z direction) the image lens and the lensed panel image appear larger in the viewing plane, and the side walls of the pyramid tilt more to accommodate the reduced space allotted to the panel images. In a specific embodiment where the image is the text of a document, the image lens might be referred to as a "document lens", however an image lens or document lens will work equally well with other images, such as maps, blueprints, and large engineering drawings.

20 The detail required in the lensed panel image and the context-preservation and compression of the side panel images are automatically provided by the 3D transformations of the full image. To do the transformations, the dimensions of the full image, the truncated pyramid, the viewing plane, and viewpoint are all described within the display system by coordinates in a common 3D space. The particular coordinates used are determined by the relative resolution of the lensed panel image and the global image, where the lens is located on the full image, and how the user manipulates user controls provided to adjust these parameters. The viewpoint is a point above the truncated top of the pyramid, but in some embodiments, the viewpoint moves around based on the movement of the image lens in order to keep the lensed panel in view.

30 In an interactive embodiment, a user manipulates an input device, such as a mouse or a keyboard, to move the image lens over the global image and to increase the magnification of the lensed panel, which are the visual equivalents of sliding the image lens around the image and bringing the image lens closer to the user. The interactive embodiment might also provide a means for adjusting the size of the image lens in much the same way as a window is resized.

35 The response time for the user commands in the interactive environment is quick, since the transformations are simple linear transformations. If available, the transformations of the image from 2D to 3D and back are performed with hardware especially suited to manipulating 3D graphics images. The system is also made quicker by generating and combining all the necessary transforms, and then rendering the objects of the full image through the combined transform. Where the objects are text, further improvements are possible using greeked text for the side panels.

Image lenses can also be nested.

40 A further understanding of the nature and advantages of the inventions herein may be realized by reference to the remaining portions of the specification and the attached drawings, in which:

Fig. 1 is an illustration of how magnification, without more, obscures portions of an image;

Fig. 2 is an illustration of an image display system including an image processor;

Fig. 3 is a block diagram of the image processor shown in Fig. 2;

Fig. 4 is an illustration of the effects of an image lens on an image;

45 Fig. 5(a) is an illustration of the division of an image into panels;

Fig. 5(b) is an illustration of the panels after being transformed;

Fig. 6 is a flowchart describing a process of displaying an image using an image lens;

Fig. 7(a) is a flowchart describing a process of rendering a lens panel;

Fig. 7(b) is a flowchart describing a process of rendering a side panel;

50 Fig. 8 is a displayed image of a multi-page document as seen through an image lens with text in the side panels greeked; and

Fig. 9 is a line diagram of a specific embodiment of the present invention wherein an nested image lens is used on the image provided in the image lens panel of a nesting image lens.

55 Fig. 2 is an illustration of a display system 100 according to the present invention which is used to display images using image lenses, and allows a user to manipulate parameters controlling the display of the images. Display system 100 comprises a display 102 which has a display surface 104, an image processor 106, a mouse 108 with an interface 142 to processor 106, and a keyboard 110 with an interface 144 to processor 106. An interface 140 between display 102 and processor 106 is also shown. In a particular embodiment, dis-

play 102 is a 1280 x 1024 color monitor manufactured by Silicon Graphics, and image processor 106 is a Silicon Graphics Iris workstation, model 4D/420 GTX. In alternate embodiments, mouse 108 and/or keyboard 110 are replaced by other input devices.

Fig. 3 is a block diagram showing processor 106 in further detail. Processor 106 comprises a central processing unit (CPU) 130, image storage 132 which stores a logical representation of an image to be displayed (full image), a memory 134 for storage of variables and intermediate results of CPU 130, and a program storage 136 which contains CPU instructions which implement functions of display system 100. In some embodiments, processor 106 includes a 3D graphics engine 138. Fig. 3 also shows interface 140 (display output), interface 142 (mouse input), and interface 144 (keyboard input).

When display system 100 is operational, CPU 130 retrieves a full image from image storage 132. The retrieved image is either an image of objects mapped onto a 2D surface, such as pixels of various colors, or the retrieved image is a logical representation of an image to appear on the 2D surface, such as text characters in a document with an indication of the font and position in which the characters are to appear.

In either case, CPU 130, upon retrieving the image, performs the necessary processing such that the position, size and orientation of the objects in the image relative to a 2D original image surface are known. This representation of the image is the "full" image, which contains all the information content of the image, including context information. The full image can be represented in various forms, such as a bitmap at a specified resolution, or by an object representation such as a text file or listing of objects with positions and descriptions, as are well known in the art. If the full image is displayable on the display surface, then the image lens is not really needed. Nonetheless, there may be applications where an image lens is still needed when a full image is displayed with its full resolution, such as looking at close-ups of characters in a page-layout system.

After retrieving the image, processor 106 then transforms the image according to parameters specified by the user, from a 2D full image onto a 3D truncated pyramid, and then onto a viewing plane which represents display surface 104.

Fig. 4 illustrates, in general, the transformations performed on a full image. Fig. 4(a) is a 2D view of a full image 200 before being transformed. Image 200 is shown with lines 201 which are used to illustrate the effects of the transformations. By convention, the full image is mapped onto a constant z-plane in an (x, y, z) coordinate system. A smaller box, indicating a lensed panel 202 for which full detail is desired is also shown. This box is alternately referred to as the "image lens".

Fig. 4(b) is a 3D perspective view of a truncated pyramid 203 onto which full image 200 is mapped (lines 201 are left out for the sake of clarity). A viewing plane 214 relative to a viewpoint V is also shown. The five visible faces of truncated pyramid 203 are a bottom face 204, a top face 206, a left face 208, a right face 210, and a truncated lens face 212. For the 3D transformation, image 200 is divided into five "panels", or subimages, which are mapped onto the five visible faces of truncated pyramid 203. Viewing plane 214 defines what is visible from viewpoint V, and determines the eventual image to be displayed on display surface 104. To preserve context information, the movement of the image lens face 212 can be limited so that panel images are not unduly compressed.

Fig. 4(c) is a 2D view of the transformed projection onto viewing plane 214. The projections 204', 206', 208', 210', 212' are projections of like numbered faces shown in Fig. 4(b). Significantly, lines in the image lens panel (face 212') and in the four side panel images (204', 206', 208', 210') remain as lines, and lines which connect in full image 200 remain connected across panel boundaries.

The parameters of the transformations are derived from a description of viewpoint V, the boundaries of full image 200 (which are also the edges of the base of truncated pyramid 203 in most cases), the boundaries of image lens 212 and the relative distances between full image 200 at the base of truncated pyramid 203, image lens 212, view plane 214, and viewpoint V. These parameters can be adjusted by a user to have the visual effects of moving the image lens in and out (zoom) and moving the image lens over full image 200 (pan). In some embodiments, the position of viewpoint V is modified by the user or is automatically modified so as to keep the projection of image lens 212 within the bounds of viewing plane 214. In Fig. 4(b), the projection of truncated pyramid 203 occupies all of viewing surface 214, so as to efficiently use display surface 104, however parameters might be provided so that view plane 214 and/or viewpoint V are movable such that this is not the case.

Figs. 5(a)-(b) and the flowcharts of Figs. 6-7 describe the operation of a specific embodiment of display system 100 according to the present invention. Figs. 5(a)-(b) are mappings of an original full image 250 and a transformed, or "lensed", image 252. The flowcharts in Figs. 6-7 describe the instructions which are stored in program storage 136 and are used by CPU 130 to effect the transformation of image 250 into image 252. In images 250 and 252, like points are similarly labelled. For example, point A is the same point in both images, and point E in image 250 is transformed to point E' in image 252. Image 250 is essentially a 2D image placed in a 3D space, while image 252 is a 3D image (although only a projection is displayable in Fig 5(b) and on

display surface 104). In the following description, the locations of the points of Figs 5(a)-(b) in the 3D space are shown in Table 1.

$A = (0, 0, 0)$
 $B = (x_max, 0, 0)$
 $C = (x_max, y_max, 0)$
 $D = (0, y_max, 0)$
 $E = (lens_x, lens_y, 0)$
 $F = (lens_x + lens_width, lens_y, 0)$
 $G = (lens_x + lens_width, lens_y + lens_height, 0)$
 $H = (lens_x, lens_y + lens_height, 0)$
 $E' = (lens_x, lens_y, lens_z)$
 $F' = (lens_x + lens_width, lens_y, lens_z)$
 $G' = (lens_x + lens_width, lens_y + lens_height, lens_z)$
 $H' = (lens_x, lens_y + lens_height, lens_z)$

Table 1. Coordinates of Panel Vertices

By convention, the x-axis extends to the right of the display, the y-axis extends up the display, and the z-axis extends out of the display towards the viewer. The variables used to described points in the 3D space can be fixed, but in a flexible system they would be stored in memory 134. In one system, $x_max = 144.0$ display units and $y_max = 108.0$ display units, thereby providing an aspect ratio of 4/3, which is the aspect ratio of most computer monitors. In that specific system, the viewpoint (eye_x, eye_y, eye_z) is initially (72.0, 54.0, 180.0), which is centered over the image, and the lens is centered on and flush with the image, with $lens_center_x = lens_x + lens_width/2 = 72.0$, $lens_center_y = lens_y + lens_height/2 = 54.0$, and $lens_z = 0$. Starting values for $lens_width$ and $lens_height$ are 20.0 and 25.0 display units respectively.

Typically, the viewer of display 102 will manipulate mouse 108 with mouse button 109 pressed to modify the values of $lens_x$ and $lens_y$, will manipulate mouse 108 with mouse button 109 pressed while the mouse is pointed at an edge of the lens panel to modify the values of $lens_width$ and $lens_height$, and will use the Alt key 120 of keyboard 110 to increase $lens_z$ and the space bar 122 of keyboard 110 to decrease $lens_z$. Of course, for more natural movement, more than one value of $lens_x, lens_y, lens_z, lens_width$ and $lens_height$ may be modified for a given mouse command.

To provide smooth movement when the lens is pulled in and out, a logarithmic approach function is used rather than moving the lens a fixed distance in the z direction for each press of a key on keyboard 110. With the logarithmic approach, one press of the space bar closes a percentage of the distance between the image lens and the plane $z = 0$, while one press of the Alt key closes a percentage of the distance between the image lens and a z slightly less than eye_z . The image lens is limited to a maximum z of slightly less than the viewpoint to avoid total loss of context information if the image lens were to move too close to the $z = eye_z$ plane. The percentage of the distance closed is settable by the user, or it can be fixed at 10%.

Fig. 6 is a flowchart describing the high-level functions embodied in the CPU instructions stored in program storage 136. These functions are performed by CPU 130 to convert a full image into a focus + context representation. Fig. 7(a)-(b) are more detailed flowcharts of the rendering steps shown in Fig. 6. Unless otherwise indicated, the flow of the program is in block number order.

At block 300, CPU 130 identifies where the objects of the full image appear. If the image is that of a multi-page document, then this step might include parsing the document into lines and pages, determining how many pages are needed, and placing the pages in a two-dimensional page array. In the 3D space, the image begins flat on the $z = 0$ plane, bounded by the points $(x,y,z) = (0,0,0)$, $(x,y,z) = (x_max, 0, 0)$, $(x,y,z) = (x_max, y_max, 0)$ and $(x,y,z) = (0, y_max, 0)$. The actual image is not necessarily rendered at this point, but this step is useful in a document for identifying how many pages will be needed, which is useful to know in a later culling process.

At block 301, the image lens is placed relative to the image in its default position. At block 302, the boun-

daries of the five panels are determined. As Fig. 5(a) shows, the boundaries are determinable from x_{\max} , y_{\max} , lens_x , lens_y , lens_width and lens_height or from the expressions for points A, B, C, D, E, F, G and H (see Table 1).

At block 303, CPU 130 calculates the transforms of each of the panels and renders the transformed image onto the display. Because the display is a 2D perspective of a 3D truncated pyramid onto which the image is placed, the lens panel will show a portion of the full image in detail, and the side panels will show the remainder of the image in varying degrees of detail, with more detail closer to the lens panel, and the side panels will also show global context.

To perform transformation and rendering efficiently, every point of the image is not transformed, but instead only the vertices and an identity matrix are transformed. CPU 130 uses the locations of the transformed vertices to determine the extent of a panel on the display surface, and uses the transformed matrix to place points of the full image onto the display surface. In some embodiments, the transformations and rendering are performed in whole or part by the specialized 3D graphics engine 138, which is usually implemented with a dedicated integrated circuit. One example of a 3D graphics engine is the Silicon Graphics Iris 4D/420 VGX.

Because the image is transformed onto a 3D truncated pyramid which is then transformed through a perspective transform where the lens panel is the top of the truncated pyramid, the lens panel appears larger to the viewer, and if lens_z is sufficiently large relative to eye_z , the resolution of the display and the image, the details of the image portion in the lens panel are discernable.

At block 304, CPU 130 adds side panel borders from the vertices of the lens to the edges of the image. If the lens is close to the $z=0$ plane (i.e., $\text{lens_z} < \text{vis_dist}$, for some threshold value vis_dist , indicating a small "magnification" effect), the borders between side panels are not shown. In some embodiments, one frame buffer holds what the viewer currently sees, and another frame buffer is used to hold the display generated by the above process. Once the second frame buffer is completed, the two frame buffers are swapped at block 305 before CPU 130 returns to block 302.

At block 306, CPU 130 exits the program if some predetermined exit condition is met, otherwise it proceeds to block 307. At block 307, CPU 130 checks for a lens movement command, and in embodiments where it is allowed, checks for a viewpoint movement command. If a movement command has not been entered, CPU 130 moves back to block 306, otherwise it proceeds to block 308.

At block 308, CPU 130 recalculates the lens position and size. As explained above, the position adjustments modify the values of lens_x , lens_y , and lens_z , while size adjustments modify the values of lens_width and lens_height .

At block 309, the viewpoint is adjusted if necessary. As can be seen from Fig. 4(b), if lens panel 212 is positioned high enough and far enough to one side, it will move outside the pyramid defined by viewpoint V and the base image 200, in which case the lens panel will not be visible. To avoid this situation, in some embodiments, the viewpoint moves with the lens. CPU 130 generates this effect by coupling the values of eye_x and eye_y to lens_center_x and lens_center_y , so that the lens panel tends to remain visible. The position of the viewpoint affects what the user sees since, as explained above, the viewpoint is used in the calculation of the perspective transform.

In a specific embodiment, the viewpoint is moved differently depending on where the lens panel is and how it is moved. For an X-Y movement, the tracking depends on the value of lens_z relative to eye_z . If lens_z is close to zero, the lens panel is close to the $z = 0$ plane. Tracking is not really necessary in this case because, as Fig. 4(b) shows, the lens panel is much less likely to be moved outside the view of view plane 214, so viewpoint tracking is not done for lens_z below some threshold value. On the other hand, if the lens panel is close to the viewpoint (lens_z near eye_z), the viewpoint tracks the lens panel to keep the lens panel in view. This can be done by moving eye_x and eye_y according to the following: $\text{eye_x} = \text{eye_x} + (x_diff * (\text{lens_z} / \text{eye_z}))$ and $\text{eye_y} = \text{eye_y} + (y_diff * (\text{lens_z} / \text{eye_z}))$, where x_diff and y_diff are the change in the x-y coordinates of the lens.

When the lens panel is moved in the z direction, the viewpoint is moved to give the appearance that the lens is staying in one place, only growing or shrinking in size. This is done by adjusting eye_x and eye_y according to the following: $\text{eye_x} = \text{eye_x} - (x_distance * \text{lens_adj_factor})$ and $\text{eye_y} = \text{eye_y} - (y_distance * \text{lens_adj_factor})$, where lens_adj_factor is the percentage change in the value of eye_z . When the lens is moving away from the viewpoint in the z direction, $x_distance = (\text{eye_x} - \text{eye_x}_0)$ and $y_distance = (\text{eye_y} - \text{eye_y}_0)$, where $(\text{eye_x}_0, \text{eye_y}_0)$ are the initial x-y coordinates of the viewpoint centered over the image. The result is that the lens appears to stay in approximately the same position on the screen, only growing smaller. When the lens is moving toward the viewpoint, $x_distance = (\text{eye_x} - \text{lens_center_x})$ and $y_distance = (\text{eye_y} - \text{lens_center_y})$. The result is this case is that the lens appears to stay in approximately the same position on the screen, only growing larger.

Once the lens and viewpoint positions have been adjusted, CPU 130 returns to block 302 and repeats the

process.

The 3D transform for the lens panel is a translation of the panel from $z = 0$ to $z = \text{lens_z}$, followed by the 3D perspective transform. Fig. 7(a) is a flowchart of a program executed by CPU 130 to render the lens panel onto display surface 104, as might be called in block 303 of Fig. 6.

5 First, at block 350, the lens panel is translated to $z = \text{lens_z}$. Next, at block 352, clipping planes are set for the lens panel. Four clipping planes are used for a rectangular lens to indicate which portion of the full image is within the lens and which portions are not. These clipping planes are used for two purposes. In the next block, block 354, objects which are not part of the image portion within the lens panel are culled from consideration during rendering, so that they do not need to be processed. Objects which are partially in a panel, such as a
10 page 400 shown in Fig 8 which is only partially in a lens panel 402, is not culled, but part of the rendered image of that page is clipped by the clipping planes. Clipping planes are not required for clipping but are typically used where the display system has hardware that supports clipping planes, which makes them fast and make their coding simple.

At block 356, the detail image within the lens panel is rendered. The image is rendered as is known in the
15 art from the full image, the bounds of the lens panel on the full image, the bounds of the lens on the display surface, the pixel resolution of the display surface and the transformation matrix between points on the full image and points on the display surface.

Once rendered, the lens panel is displayed (or stored to a frame buffer until the rest of the display is rendered). Next, at block 358, the lens border is added to the displayed image. The lens border could be either a
20 solid line or a shaded line.

The 3D transform for a side panel of the image is a rotation about the side panel edge which forms an edge of the full image, followed by a scaling of the side panel towards the lens panel, so that the side panel meets the lens panel. After the 3D transform, the 3D perspective transform is done, however all these transforms might be done by a mere matrix multiplication.

25 Fig. 7(b) is a flowchart of a program executed by CPU 130 to render a side panel onto a display surface. At block 370, the side panel is rotated by $\tan^{-1}(\text{lens_z} / \text{panel_height})$ around the image edge which forms one edge of the side panel, where panel_height is the distance between the axis of rotation and the opposite edge of the side panel, the edge which meets with the edge of the lens panel in the full image.

Next, at block 372, CPU 130 expands the side panel so that it will again "meet" the edge of the lens panel,
30 which has been moved up in the 3D space. The side panel is expanded by a factor of $(\text{panel_height}^2 + \text{lens_z}^2)^{1/2} / \text{panel_height}$. The values of panel_height for the bottom, top, left and right side panels shown in Fig. 5(a) are lens_y , $\text{y_max} - \text{lens_y} - \text{lens_height}$, lens_x and $\text{x_max} - \text{lens_x} - \text{lens_width}$, respectively.

After rotation and expansion, the side panels "meet" the lens panel in the 3D space, so the 3D perspective transform of the side panels, when added to the transform of the lens panel show the entire image, although
35 with varying levels of detail. Since the entire image is shown, context is preserved.

At block 374, clipping planes are set for the side panel. Four clipping planes are also used here, but they form a trapezoid instead of a rectangle as was the case with the clipping planes for the lens panel. At block 376, objects entirely outside the clipping planes are culled.

At block 378, the image within the side panel being processed is rendered, using the transformation matrix
40 for the combination of the two 3D transforms and the 3D perspective transform, the side panel is then displayed or stored to a frame buffer until the rest of the display is rendered.

Fig 8 shows an display surface resulting from the application of a document lens to an image comprising multiple pages of text. As shown in Fig. 8, where the image is text, the text within the lens panel is rendered in the font for that text, but the text within the side panels may be rendered in a greeked font. In a greeked font,
45 a line of text is rendered as a single line. Greeking is used where speed is important, such as where the lens is moving and many image frames must be generated quickly to provide the appearance of motion. In one embodiment, the text of the side panels is always greeked, while in another embodiment, the text in the side panels is greeked only when the lens is in motion. Fig. 8 illustrates how global context is preserved using a display system according to the present invention. With a multi-color display, or other means for contrasting text, the
50 image displayed might include text of a different color to show occurrences of a search term. The global results of the search could be shown even where the side panels are greeked, with the greeked lines maintaining the color of the underlying text. In this way, the occurrences of search terms would still be shown for the entire image, even if the side panels were greeked.

Fig. 9 is a line diagram of a nested image lens. To produce the image in Fig. 9, first the image bounded
55 by a border 500 is transformed using an image lens as described above, resulting in side panels 502, 504, 506, 508, and a lens panel 510. Lens panel 510 is then transformed as if it were an image itself, resulting in side panels 512, 514, 516, 518, and nested image lens 520. Using a prearranged set of input commands, the viewer manipulates the movement of lens panel 510 and lens panel 520. One possibility is for a key on the

keyboard to be designated the lens pointer key which, when pressed, alters the "current" lens panel to which lens movement and sizing commands are applied. A key could be provided to turn off nested image lens 520, or the viewer could simply manipulate it to be on the surface of nesting image lens 510 for the same effect.

5 The image in Fig. 9 would be displayed on nine panels. However, it should be apparent that the nesting is not limited to two levels, but could be as many as is useful for the particular image. Additionally, it should also be apparent that the nested image lens need not only modify the nesting image lens, but the nested image lens could modify a side panel of the nesting image. It should also be apparent that the viewpoint tracking could be modified such that the viewpoint tracks lens panel 510 to keep it in view and tracks lens panel 520 in view relative to image 500 or relative to the image on lens panel 510. The former would be an extension of
10 the viewpoint tracking described above with lens panel 510 used as a base, whereas the latter would use lens panel 500 as a base.

The above description is illustrative and not restrictive. Many variations of the invention will become apparent to those of skill in the art upon review of this disclosure. The scope of the invention should, therefore, be determined not with reference to the above description, but instead should be determined with reference
15 to the appended claims along with their full scope of equivalents.

Claims

- 20 1. A method for displaying an image on a display surface, preserving context and detail information even when the image contains too much detail to be displayed all at once on the display surface, comprising the steps of:
 - storing a data block representing objects appearing in the image, wherein a full image is an image generated from said data block without loss of information;
 - 25 determining layout of said full image in two dimensions which is to be displayed on the display surface and includes context information;
 - dividing said layout into a lens panel which maps to portions of said full image containing desired detail information, and dividing a remainder of said layout not within said lens panel into a plurality of side panels;
 - 30 mapping said lens panel and said plurality of side panels onto a truncated pyramid in a 3D space, said lens panel forming a truncated top surface of said truncated pyramid and said side panels forming wall surfaces of said truncated pyramid;
 - transforming said truncated pyramid through a perspective transform to form a 3D transformed layout, where a viewpoint for said perspective transform is above said lens panel; and
 - 35 rendering said full image onto the display surface using said 3D transformed layout to form a rendered image, wherein said perspective transform provides a detailed view of said lens panel and context views of said plurality of side panels.
2. The method of claim 1, wherein (1) the objects are text characters, lines of characters, and pages of lines, and the image is a document, or (2) the objects are nodes and lines, and the image is a map.
- 40 3. The method of claim 1 or 2, wherein said step of mapping onto a 3D truncated pyramid comprises the steps of:
 - determining a magnification factor of image portions under said image lens relative to bounds of said full image;
 - 45 setting a height of said truncated pyramid such that a ratio of a distance from said viewpoint to a base of said truncated pyramid and a distance from said viewpoint to said top face of said truncated pyramid equal to said magnification factor; and
 - rotating and expanding said plurality of side panels to fit said walls of said truncated pyramid.
- 50 4. The method of claim 1, 2 or 3, wherein said plurality of side panels are rendered in a lower resolution than the lens panel.
5. The method of any of the preceding claims, wherein the image is a text documents and said lower resolution is provided by using a greeked font for text in said plurality of side panels.
- 55 6. A context-preserving image display apparatus which displays an image which cannot be displayed on a display surface without some loss of information, comprising:

image input means for accepting a pixel array representing the image;
panelling means, coupled to said image input means, for mapping pixels in said pixel array to panels in a set of panels;

5 transformation means, coupled to receive from said panelling means pixels assigned to panels of said set of panels, for transforming said pixels according to a transform function for said panel to which said pixels are assigned, wherein a transform function for a top panel is a translation from a base plane to a plane between said base plane and a viewpoint and a transform function for a side panel is a rotation about a line in said base plane towards an edge of said top panel and a scaling such that edges of said top panel and said side panel which were adjacent before said transformations are adjacent after said
10 transformations, thereby forming a display pixel array; and

a display driver, coupled to received said display pixel array from said transformation means, where said display driver provides said display pixel array to the display, to allow the display to illuminate pixels of the display according to values in said display pixel array.

15 7. The apparatus of claim 6, wherein said panelling means further comprises means for moving boundaries between panels.

8. The apparatus of claim 6 or 7, wherein said transform of said top panel results in a display of substantially all information content of the image within said top panel, and wherein said set of panels comprises a top, bottom, left, and right side panel which are projections onto the display surface of a truncated pyramid onto which the image has been mapped .
20

9. A method of scaling an image, comprising the steps of:
describing the image by a set of objects located within an image bounding rectangle on a base plane;
25

locating an image lens within said image bounding rectangle, where said image lens is a rectangular area with sides parallel to said image bounding rectangle;

dividing said image bounding rectangle into a lens panel, being the portion of said image bounding rectangle within said image lens, and four side panels, a side panel being the portion of said image bounding rectangle between an edge of said image bounding rectangle and a parallel edge of said image lens nearest said edge of said image bounding rectangle and lines drawn to corners of said edges of said image bounding rectangle and said image lens, whereby each of said side panels form a trapezoid;
30

defining a lens transformation of said lens panel from an original size to a new size, said new size being larger than said original size;

defining a side transformation of said side panels from original trapezoids to new trapezoids defined by redrawing said lines drawn to corners to be drawn to corners of transformed lens panel; and
35 rendering the set of objects through said lens and side transformations onto a display surface.

10. The method of claim 9, wherein said step of rendering is different for said lens panel and said side panels, said step of rendering for a side panel being of a lower resolution than a rendering for said lens panel.
40

45

50

55

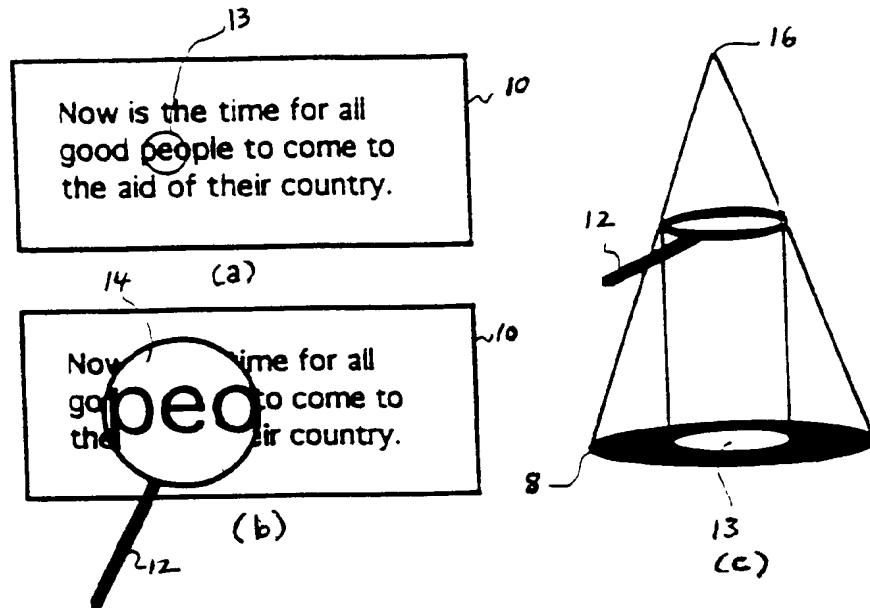


FIG. 1

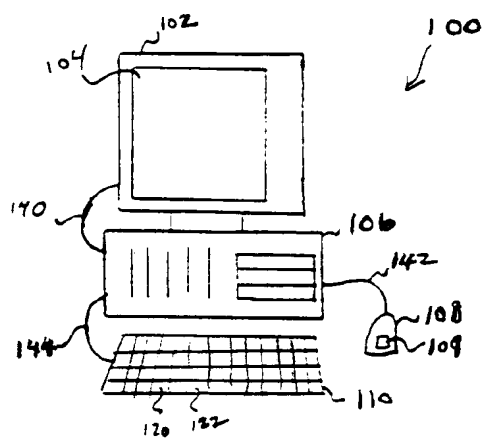


FIG. 2

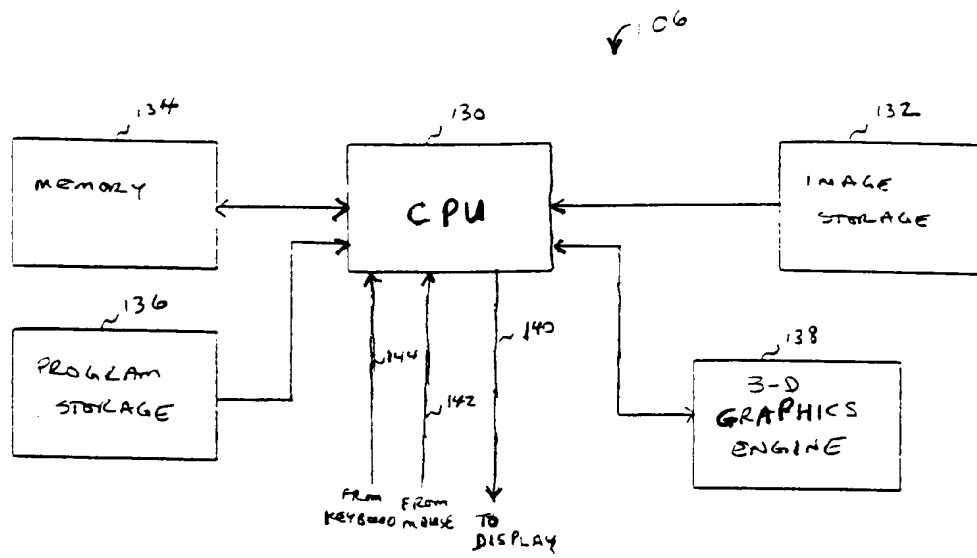


FIG. 3

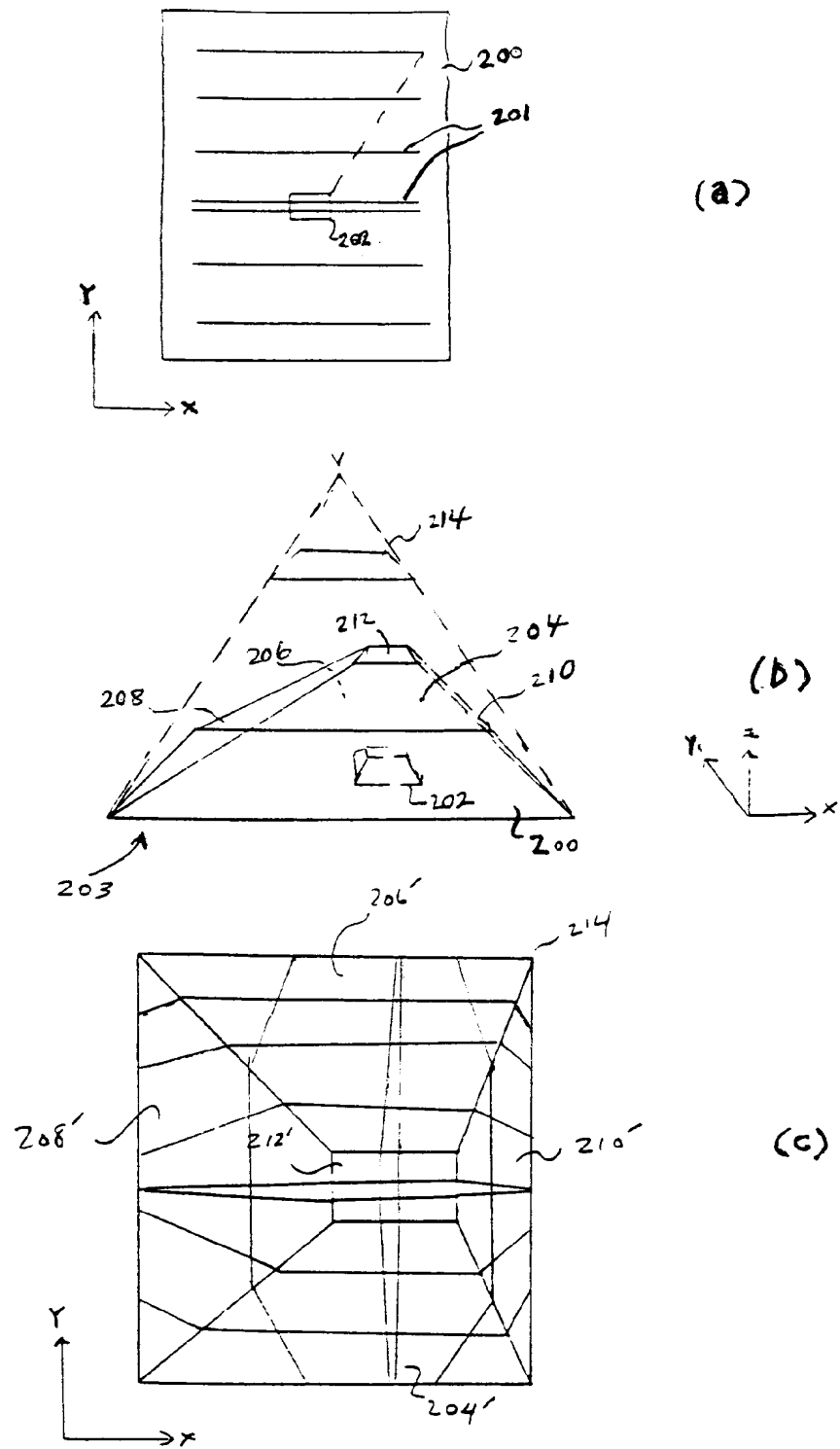
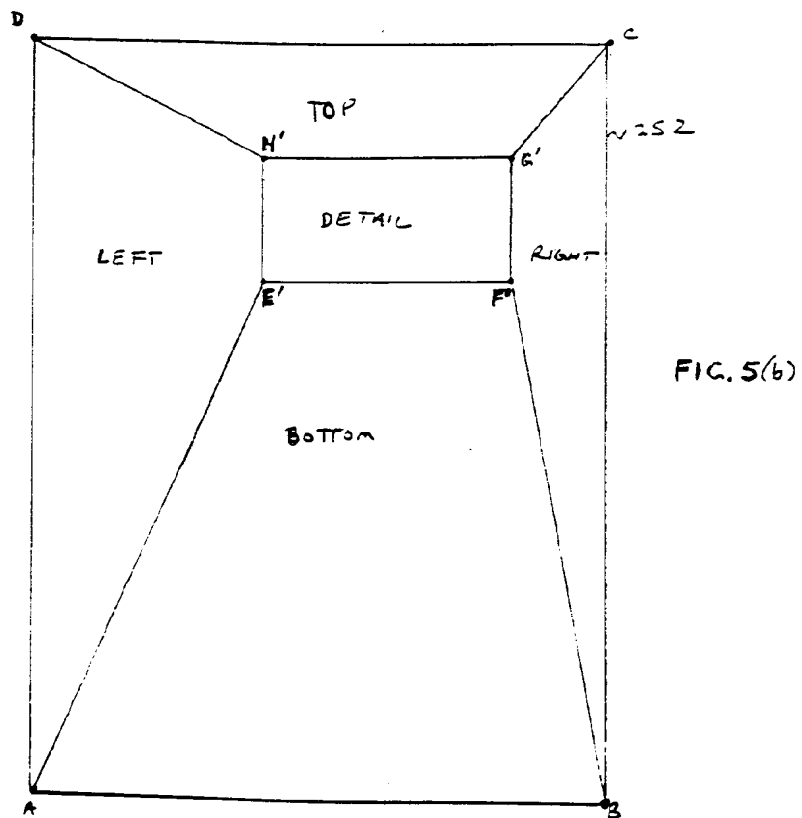
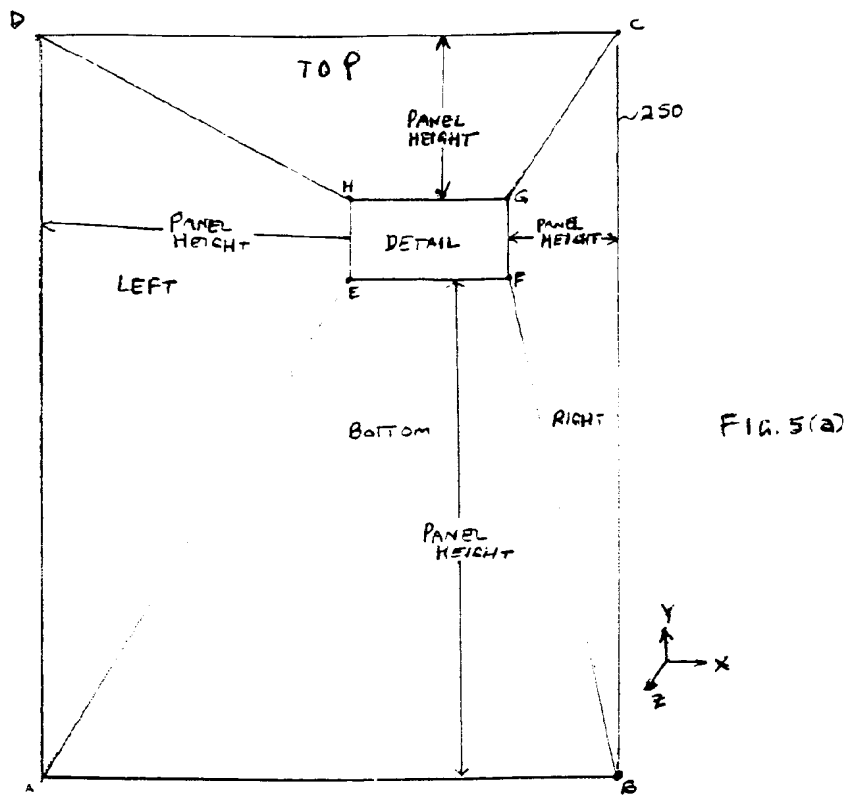


FIG. 4



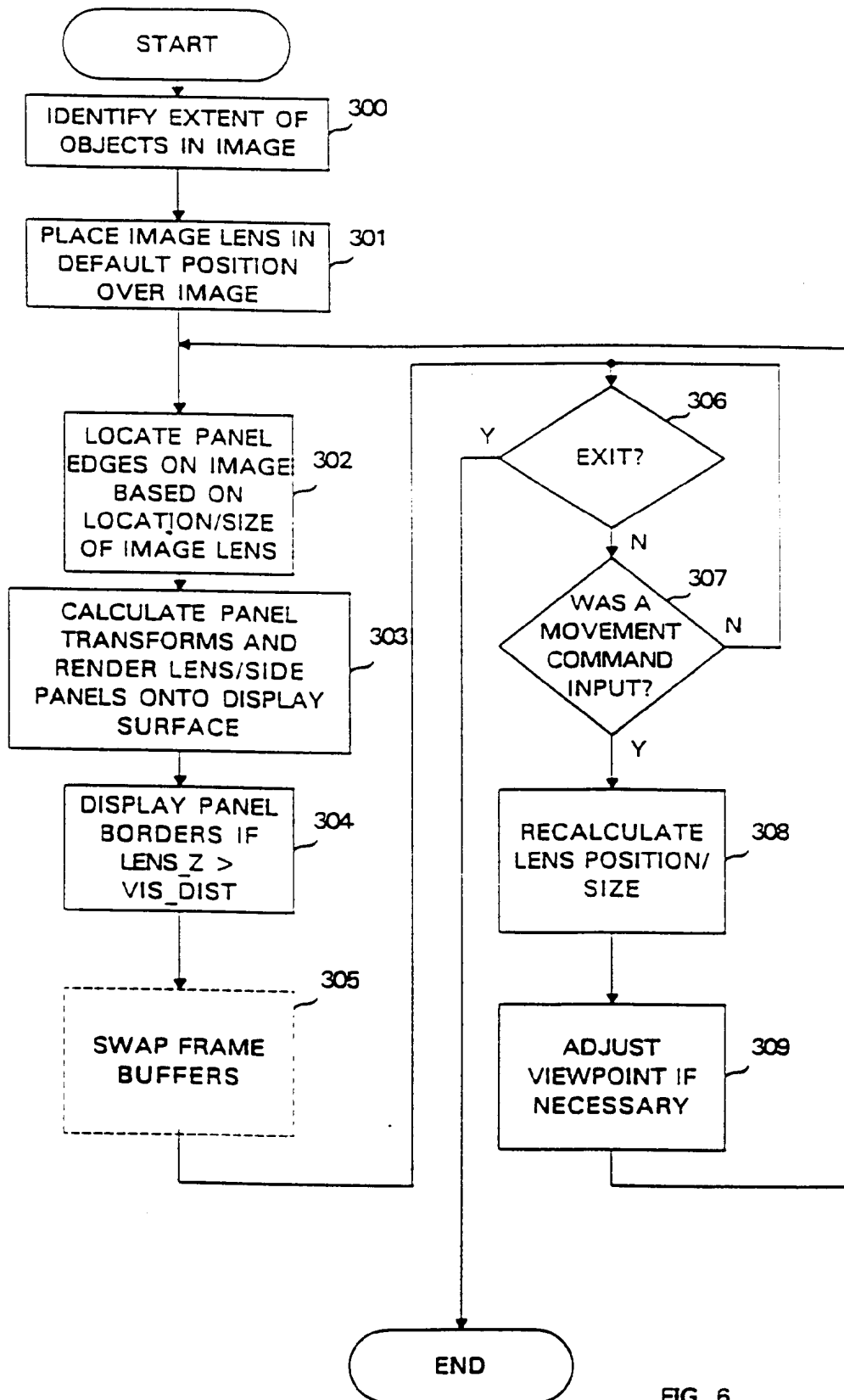


FIG. 6

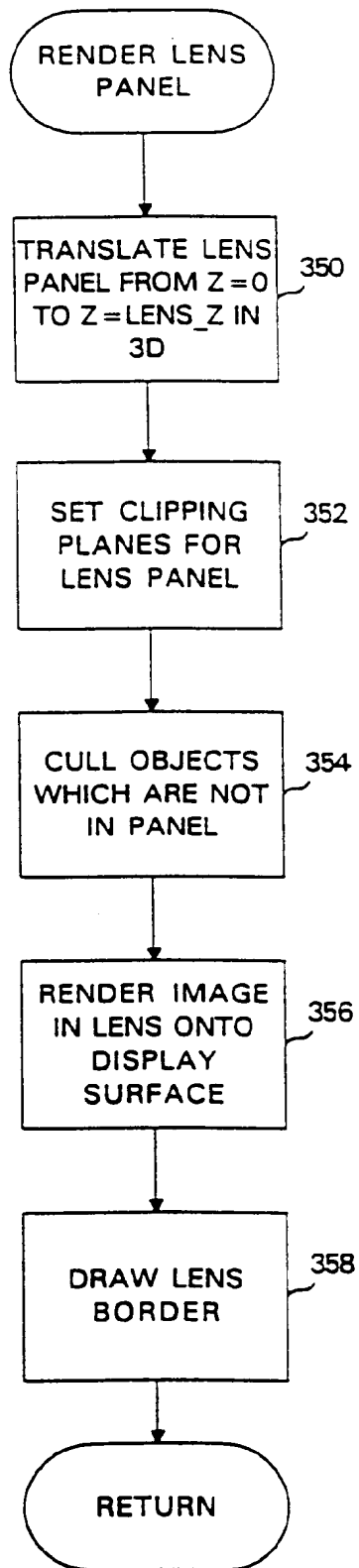


FIG. 7(A)

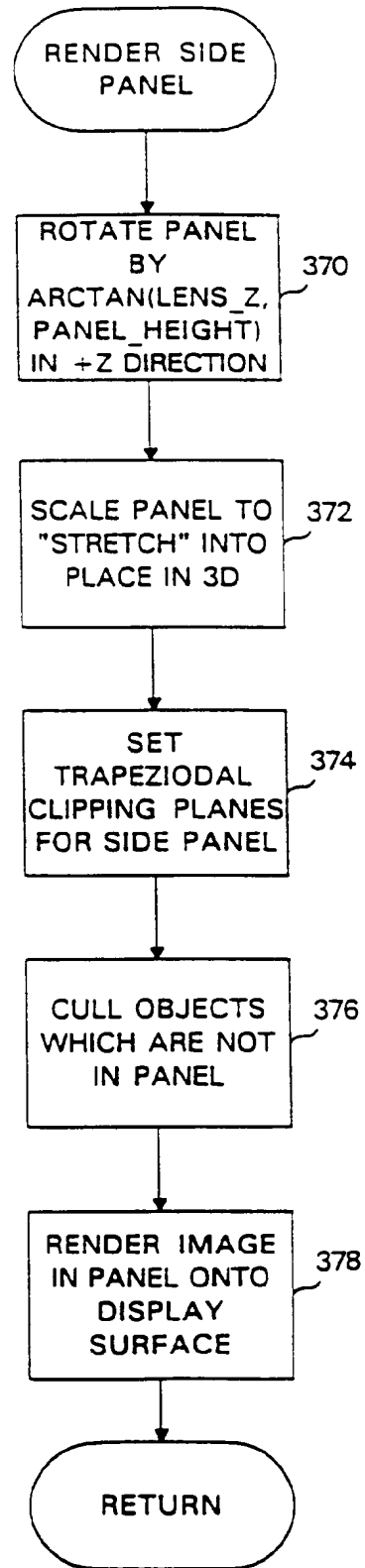
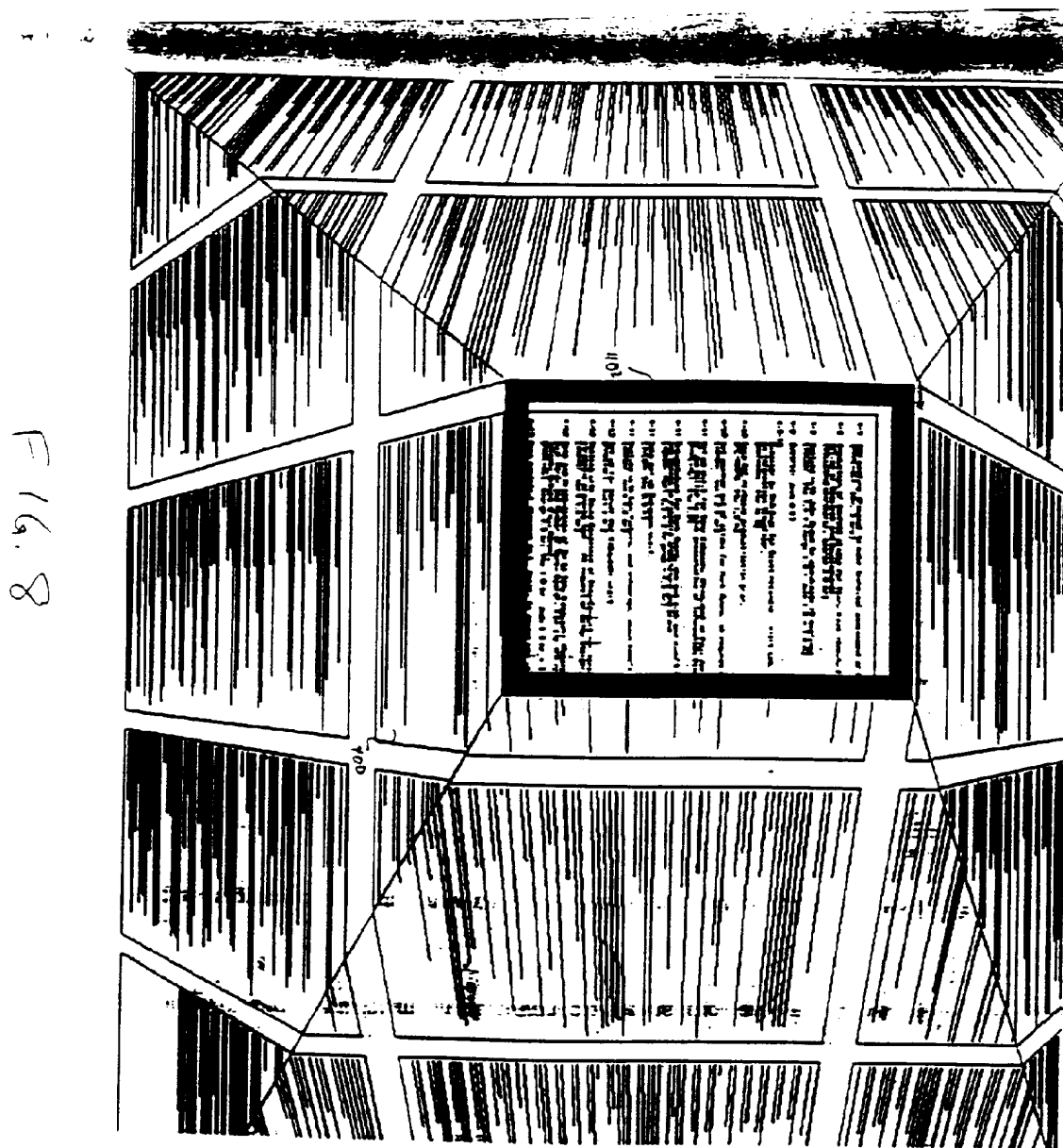


FIG. 7(B)



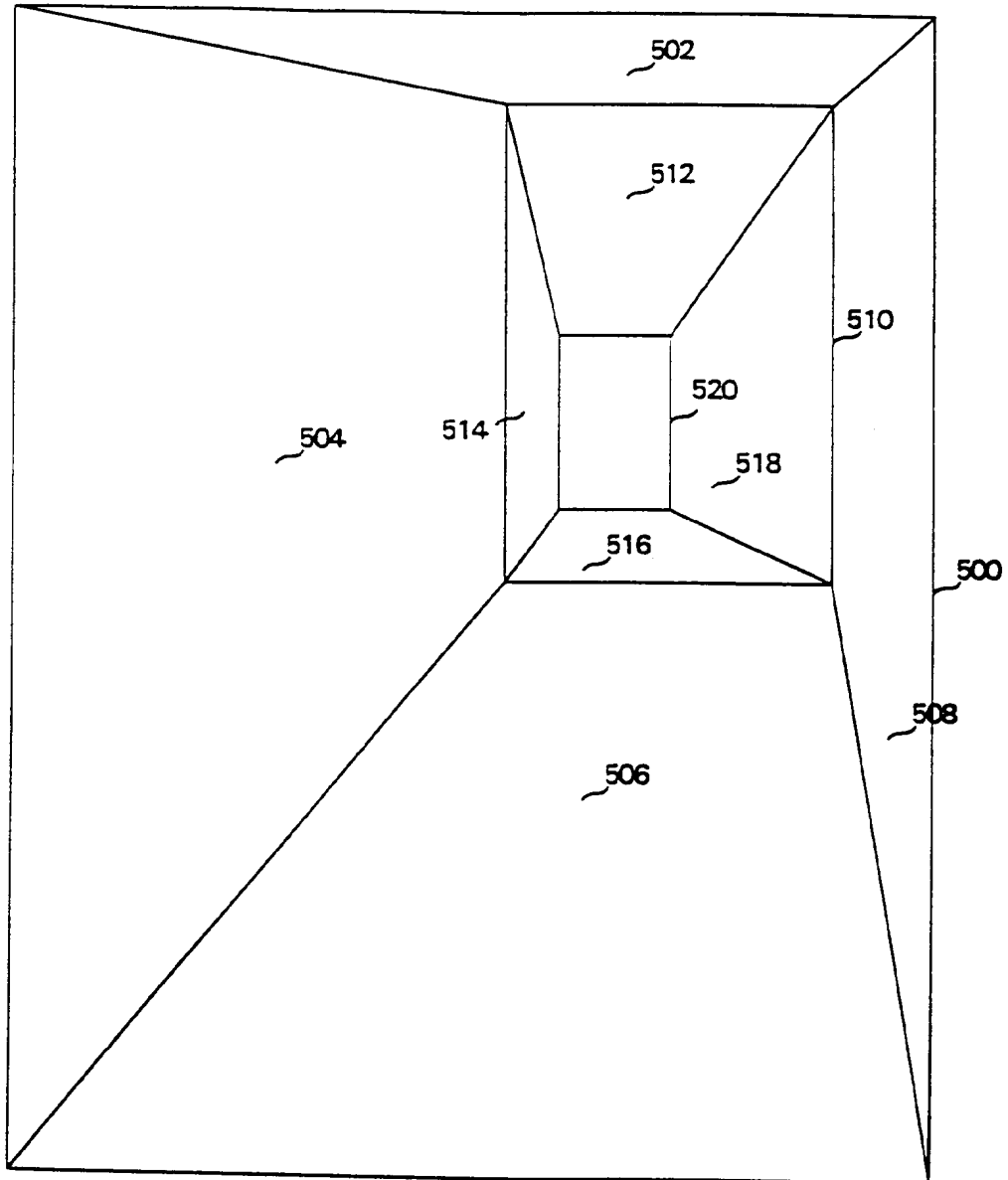


FIG. 9



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 30 7833

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
P,X	SIXTH ANNUAL SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY. PROCEEDINGS OF THE ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, PROCEEDINGS OF 6TH ACM SYMPOSIUM ON USER INTERFACE AND SOFTWARE TECHNOLOGY, ATLANTA, GA, USA, 3-5 NOV. 1993, ISBN 0-89791-628-X, 1993, NEW YORK, NY, USA, ACM PRESS, USA, PAGE(S) 101 - 108 Robertson G G et al 'The Document Lens' * the whole document * ---	1-10	G06T3/00
Y	HUMAN FACTORS IN COMPUTING SYSTEMS. REACHING THROUGH TECHNOLOGY. CHI '91. CONFERENCE PROCEEDINGS, NEW ORLEANS, LA, USA, 27 APRIL-2 MAY 1991, ISBN 0-89791-383-3, 1991, NEW YORK, NY, USA, ACM, USA, PAGE(S) 173 - 179 Mackinlay J D et al 'The Perspective Wall: detail and context smoothly integrated' * page 176, left column, paragraph 4 - right column, paragraph 1; figure 3 * ---	1-10	
Y	CHI '92 CONFERENCE PROCEEDINGS. ACM CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS. STRIKING A BALANCE, PROCEEDINGS OF 1992 CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, MONTEREY, CA, USA, 3-7 MAY 1992, ISBN 0-89791-513-5, 1992, NEW YORK, NY, USA, ACM PRESS, USA, PAGE(S) 83 - 91 Sarkar M et al 'Graphical fisheye views of graphs' * page 86, paragraph 4 - page 87, paragraph 5; figures 4,5 * --- -/--	1-10	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 6 February 1995	Examiner Pierfederici, A
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons ***** & : member of the same patent family, corresponding document</p>			

EPO FORM 1500 (03.84) (P4/C01)



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 30 7833

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	EP-A-0 399 663 (QUANTEL LTD) 28 November 1990 * abstract * * page 4, line 2 - line 6 * ---	1-10	
A	EP-A-0 447 095 (XEROX CORP) 18 September 1991 -----		
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 6 February 1995	Examiner Pierfederici, A
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons ***** & : member of the same patent family, corresponding document</p>			

EPO FORM 1503 03/92 (P04C01)